Software Defined Radio Basics
A Presentation to the Paducah Amateur Radio Association

March 10, 2008

William Parmley, KR8L

Good Evening. We're going to include a technical presentation with tonight's meeting. Since we are also having a testing session we wanted to do something fairly short, so I volunteered to give a brief talk on a project that Steve (AI4ZL) and I have been working on. Credit goes to Steve for coming up with the original idea and finding all of the resources on the Internet. It sounded like an interesting and educational project, so I decided to participate.

The subject is Software Defined Radio (SDR). All I plan to do tonight is give you a very basic overview and show you how you can learn more by building your own simple SDR receiver. For those who want to know more or who want to try the building project a list of Internet links, as well as the complete text of this presentation, is available at sdr.KR8L.us.

I think the best way to get started is to review some of the basic concepts behind conventional receivers, and then show how the SDR differs. First, let's look at one of the most basic receiver designs, the direct conversion (DC) receiver (Figure 1).

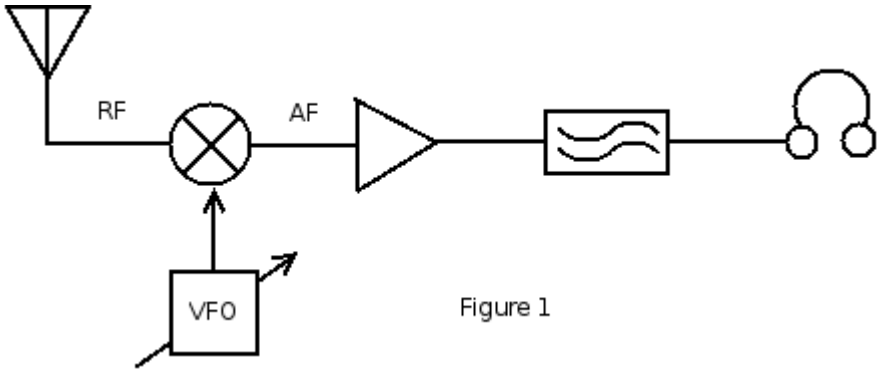


Figure 1

In the simplest version of the direct conversion receiver, radio frequency (RF) signals are picked up by the antenna and are fed to a mixer where they are combined with an RF signal from a variable frequency oscillator (VFO). The VFO operates at around the same frequency as the on-air signal so that the difference between the two signals is an audio frequency (AF). We simply amplify this audio signal, pass it through some sort of audio filter (either active or LC type) to add a degree of selectivity, and feed the result to our headphones.

Now, let's compare this with a basic superheterodyne (superhet) receiver (Figure 2).

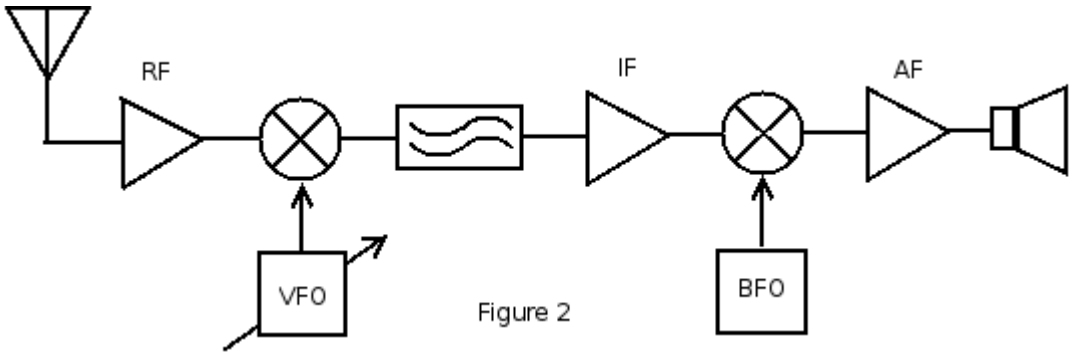


Figure 2

In a basic superhet RF signals are again picked up by the antenna. They are amplified and then fed to a mixer where they are combined with an RF signal from a VFO. In the case of the superhet the mixer product that we use is not at an audio frequency, but rather is at some new radio frequency that we call the intermediate frequency (IF). (The IF can be either higher or lower in frequency that the original RF.) We pass this signal through either a mechanical or a crystal filter to provide selectivity, amplify the IF signal, and feed it to another mixer where it is combined with an RF signal from the beat frequency oscillator (BFO). The resulting output, just as with the DC receiver, is an audio frequency that we can amplify and send to our loudspeaker.

You will notice that both the DC receiver and the superhet receiver provide the same three basic functions:

1. Tuning – the ability to listen to one particular frequency within a specific range of frequencies. This is accomplished by mixing the incoming signal with the signal from the VFO.
2. Selectivity – the ability to isolate one particular signal from all of the other signals around it. This is accomplished by using filters operating either at RF or at AF.
3. Output – this is the ultimate goal of the receiver – to convert the RF energy of the signal of interest into AF energy that the operator can sense and understand.

Now let's talk about the software defined radio (SDR) receiver (Figure 3). With the SDR all three of these functions are accomplished not by hardware, but by software running in your computer. You can imagine what a powerful thing this is! Any time you want to modify the performance of your receiver you no longer need to heat up your soldering iron and replace parts, you simply need to reprogram your computer (or download a new program from someone who knows how to do that).

So, how can we turn our computer into a radio? We know that our computers operate at very high frequencies. In fact, many modern computers have components that operate up into the microwave region. However, we really don't have any way to inject an RF signal directly into the computer so that we can process it. However, we do know how to convert RF to AF as we've just discussed. We also know how to get AF into and out of our computer, since almost all modern computers include some sort of sound card technology.

Here is a very simple version of the SDR:

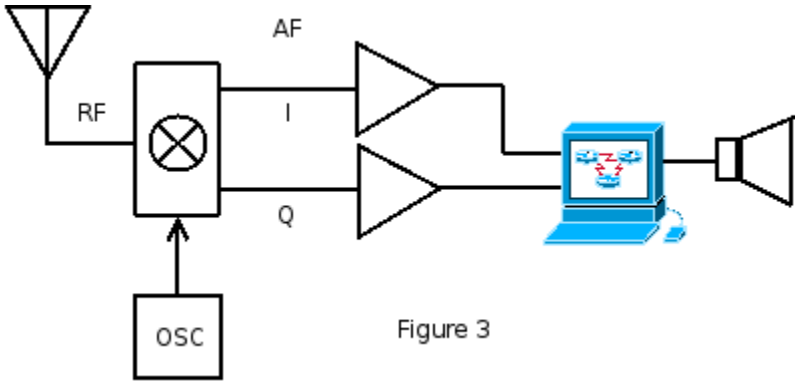
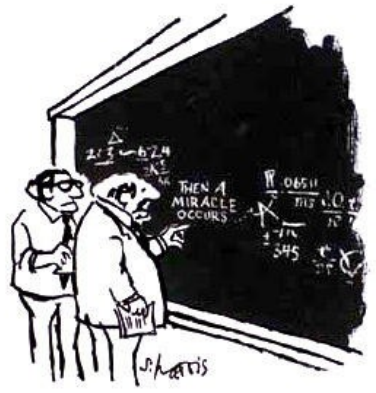


Figure 3

In some ways this looks a lot like our DC receiver. In the SDR once again the RF signals are picked up by the antenna and once again they are fed to a mixer. Note two big differences from our previous examples, however: (1) the oscillator in this case is fixed and not variable, and (2) the output of the mixer is special in that it includes not one but two audio signals. Again we amplify the audio signals, but now instead of using a filter to provide selectivity we send both audio signals to our stereo sound card, one to the left channel and one to the right channel.

A few words about these two audio signals: first, since we have not introduced any selectivity, they represent the entire range of frequencies that we want our receiver to tune (which might be, say, 14.000 to 14.100 MHz for example). Of course, that frequency range has been shifted down from RF to AF, but otherwise it is an exact copy of the original. Second, the two audio signals are identical to each other except for one very important difference: one of the signals is shifted in phase from the other by 90 degrees. We refer to these two as the "**I**" (or in-phase) and the "**Q**" (or quadrature) signals.



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

At this point the theory gets very, very difficult and involves a lot of very complex mathematics. What is important for us to understand is that our computer will perform some very complex and powerful digital signal processing (DSP) functions for us, and the source material that it needs in order to do this work is these two otherwise identical signals, separated by a 90 degree phase difference. The computer's sound card makes the necessary analog to digital (A/D) conversion of the two audio signals and then passes the digital version to our SDR software. The software performs several functions, the first of which is to present the entire spectrum of signals to the operator in the form of a graphic display (Figure 4).

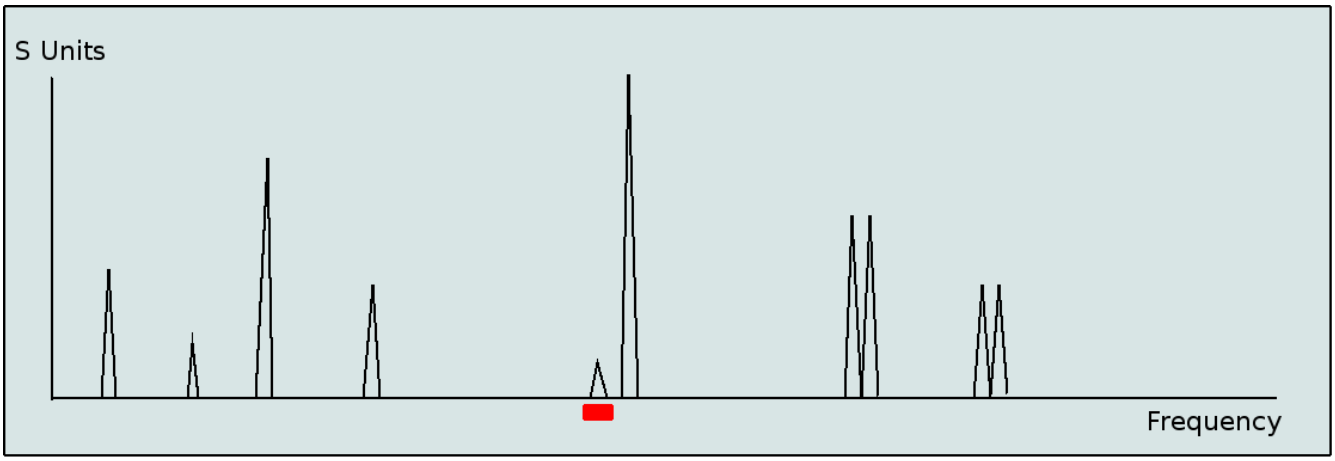


Figure 4

This enables us to operate our radio simply by clicking our mouse. The operator tunes the desired signal by clicking the mouse at the appropriate spot on the spectrum display, causing the software to demodulate the signal at that location by using appropriate mathematical functions to recover the transmitted intelligence. The software provides the necessary selectivity by using mathematical functions to filter the desired signal. The operator can engage different filter types (e.g., USB, LSB, CW, FM, RTTY) with a mouse click, and can change the filter width by clicking and dragging. Finally, the computer performs a digital to analog (D/A) conversion, again using the sound card, to output an audio signal that can be sensed and understood by the operator.

Now let's look at this little device [holding up the Softrock Lite receiver mounted on a piece of wood]. This is a simple software defined radio receiver that I built from a kit. The kit was purchased from Tony, KB9YIG, and there is a Yahoo! discussion group devoted to providing information and support for builders. This particular kit, which cost $12 (including mailing), covers the lower end of 20 meters. The exact frequency coverage depends on the sampling rate of the sound card. If you are using a 96 kHz card then it will tune from about 14.000 to 14.090 MHz, which is wide enough to tune not only CW but also RTTY and PSK-31.

You're probably wondering about my unusual choice of “cabinetry.” A small board like this would normally be mounted in a small box or maybe even an Altoids tin, but I wanted to mount it so that you can see both sides of the board and, since it is fairly delicate, I didn't want to pass around the bare circuit board. This [indicating] is the antenna connection and this [indicating] connects to the sound card. The red and black leads are for 12 volt DC.

You will notice that the board makes use of both leaded and surface mount components. I had never done any surface mount work, which is another thing that attracted me to the project since it was a cheap way to learn a new construction technique. Thanks to Steve, AI4ZL, for showing me how to do this.

There are several versions of SDR software that you can obtain to use with this receiver. All are free and all are downloadable via the Internet. The *Rocky* program is the simplest, while the *WinRad* program provides more features. The *Flex-Radio* program is the most complex and is the same software that runs the commercially available Flex-Radio, which costs several thousand dollars.

There are receiver kits available for many of the HF bands and there are also SDR transceiver kits available. Links to the Softrock Yahoo! group, kit information, documentation, and software are available at sdr.KR8L.us as well as being listed below.

Links:

Softrock Yahoo! Group:
http://groups.yahoo.com/group/softrock40/

Softrock Lite Receiver Information and Links:
http://groups.yahoo.com/group/softrock40/message/7846
http://groups.yahoo.com/group/softrock40/files/SoftRock%20v6.0%20docs/SoftRock%20Lite%20docs/

Softrock Transceiver Information and Links:
http://groups.yahoo.com/group/softrock40/message/15113
http://groups.yahoo.com/group/softrock40/files/RXTXv6.2%20files/

Flex Radio Software:
http://www.flex-radio.com/

Rocky Software:
http://www.dxatlas.com/rocky/

WinRad Software:
http://digilander.libero.it/i2phd/winrad/index.html

Firefly QRP SDR Kit (another SDR project suggested by John, KO4XJ):
http://www.qrpkits.com/firefly.html